AIAA/AAS Astrodynamics Specialist Conference
8-11 August, 2005, Lake Tahoe, CA

AAS-05-267

Adventures in Parallel Processing: Entry, Descent and Landing Simulation for the Genesis and Stardust Missions

Daniel T. Lyons
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, 91109

Prasun N. Desai
NASA Langley Research Center, Hampton, Virginia, 23681

This paper will describe the Entry, Descent and Landing simulation tradeoffs and techniques that were used to provide the Monte Carlo data required to approve entry during a critical period just before entry of the Genesis Sample Return Capsule. The same techniques will be used again when Stardust returns on January 15, 2006. Only one hour was available for the simulation which propagated 2000 dispersed entry states to the ground. Creative simulation tradeoffs combined with parallel processing were needed to provide the landing footprint statistics that were an essential part of the Go/NoGo decision that authorized release of the Sample Return Capsule a few hours before entry.

Introduction:

Both Genesis and Stardust are sample return missions. Genesis has already returned to Earth and Stardust is scheduled to return on January 15, 2006. Both spacecraft use a heatshield and a backshell to protect the sample during reentry. Both entry capsules use a drogue chute to maintain attitude stability at supersonic speeds and a main chute to control the final descent speed. This paper will describe the Entry, Descent and Landing simulation tradeoffs and techniques that were used to provide the data required for an approval to proceed with entry. This approval had to be made during the critical time following the last maneuver that targeted entry and the next maneuver that prevented the main spacecraft from reentering. Minimizing the dispersion at entry requires performing the last targeting maneuver as close to entry as practical, while allowing time to track the spacecraft after the maneuver, estimate the entry state and uncertainty, authorize the release of the entry capsule, and upload the commands to enable release and the subsequent deflection maneuver. The desire to minimize the entry dispersions meant that only one hour was available for doing the required Monte Carlo simulation which propagated 2000 dispersed entry states to the ground. Creative simulation tradeoffs combined with parallel processing were needed to provide the landing footprint statistics that were an essential part of the Go/NoGo decision that had to be made in the exciting final hours before entry.

Acronym List:

AEPL	Atmospheric Entry/Powered Landing (Computer Program for EDL analysis)
EarthLS	Earth Landing Site. (Computer Program for visualization of Landing Sites.)
EDL	Entry, Descent and Landing
MER	Mars Exploration Rover, a NASA project that landed two rovers on Mars.
NASA	National Aeronautics and Space Administration.
POST	Program to Optimize Simulated Trajectories (another Computer Program)
SRC	Sample Return Capsule (Entered the Atmosphere).
UTTR	Utah Test and Training Range (where the SRC landed.)

Mission Descriptions:

The 494 kg Genesis mission[1] was launched on August 8, 2001 on a mission to collect samples of the solar wind in high purity wafers of several different types. After orbiting L1 for several years, Genesis returned to Earth on September 8, 2004. The 206 kg Sample Return Capsule (SRC) landed in the Utah Test and Training Range (UTTR), while the main spacecraft was diverted by a propulsive maneuver to miss the Earth. Although the parachutes failed to deploy, the SRC landed very close to the targeted location. The initial despair of seeing the entry capsule embed itself in the dirt has gradually receded as good news of successful recovery of the samples from the collection wafers has been released to the public.

The 385 kg Stardust mission[2] was launched on Feb 7, 1999 on a mission to collect dust samples from the tail of comet and from interplanetary space. Stardust flew within 236 km of comet Wild-2 on January 2, 2004 and collected dust samples in high purity aerogel collectors. It also took pictures as it flew by the asteroid Anne Frank on November 2, 2002. Stardust is scheduled to return to Earth on January 15, 2006. The 46 kg sample return capsule will also land in the Utah Test and Training Range Southwest of Salt Lake City. Both Genesis and Stardust were designed and built by Lockheed Martin Astronautics in Denver, Colorado.

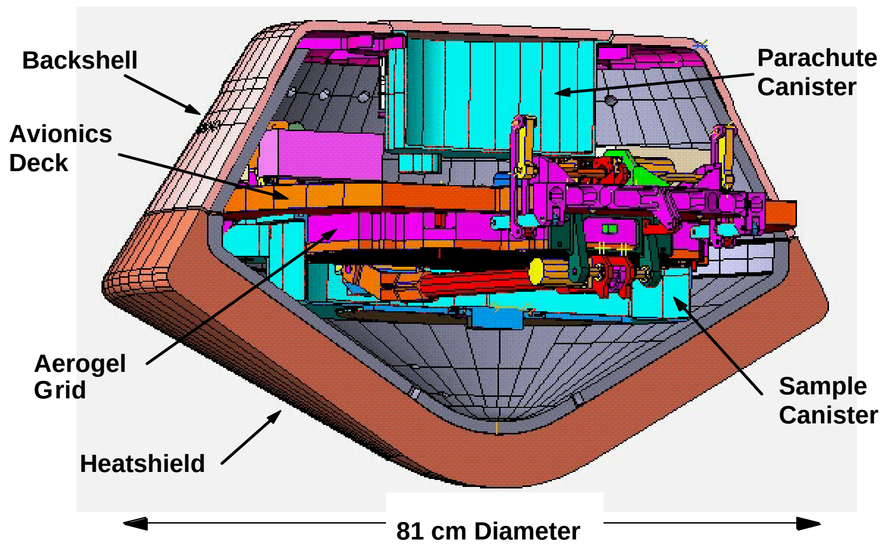

Figure 1 Stardust Sample Return Capsule (SRC)

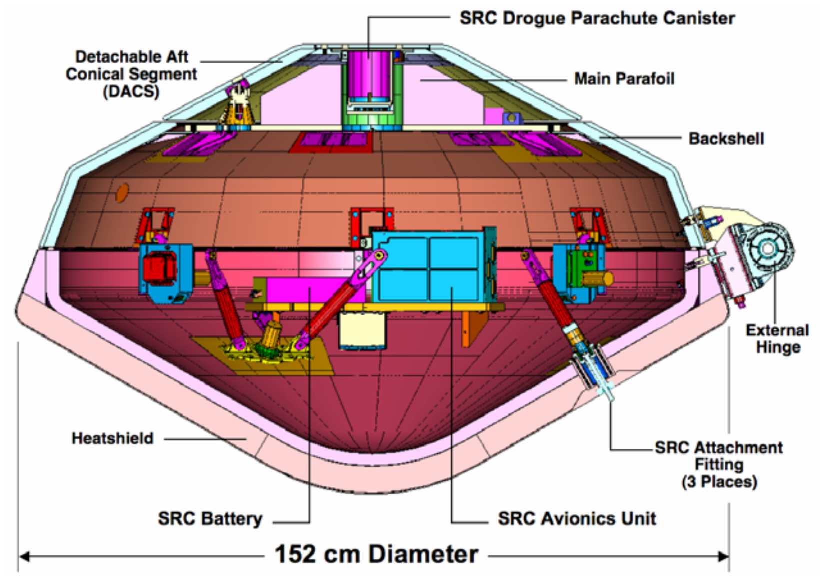
SRC Drogue Parachute Canister
Detachable Aft Conical Segment (DACS)
Main Parafoil
Backshell
External Hinge
Heatshield
SRC Attachment Fitting (3 Places)
SRC Battery
SRC Avionics Unit
152 cm Diameter

Figure 2 Genesis Sample Return Capsule (SRC)

One interesting difference between the two missions that is important for Entry, Descent and Landing (EDL) is the entry speed. Genesis never achieved escape velocity, and thus returned to Earth with a relatively low entry speed that was slightly less than escape velocity (11 km/sec). On the other hand, Stardust will reenter with a speed of 12.8 km/sec and become the fastest man-made object to enter the Earth's atmosphere. The difference in entry speed required a different material for the heat shield for Stardust in order to survive the higher heating rate at entry. The Genesis carbon-carbon heatshield did not experience significant mass loss due to ablation. The Stardust carbon-phenolic heatshield is designed to ablate, and is expected to lose about 1.15 kg during entry.

Table 1 Differences between the Stardust and Genesis Entry Capsules

	Entry Mass	SRC Area	Entry Speed	Entry FPA	Max.Decel.
Stardust	45.2 kg	0.517 m^2	12.48 km/s	-8.2°	32.6 g
Genesis	205.6 kg	1.82 m^2	10.77 km/s	-9.0°	32.2 g

After passing through the maximum deceleration and heating phases, both entry capsules were designed to use a deceleration sensor to trigger deployment of a drogue chute to maintain attitude stability until the main chute could be deployed. Genesis planned a mid-air capture during the day using a helicopter to snatch the parafoil chute. Stardust will descend all the way to the ground on a circular, disk band gap chute during a night entry.

Further information about these and other exciting NASA missions can be found at http://www.jpl.nasa.gov/

Entry, Descent and Landing Timeline:

Both missions were designed such that the Navigation team would provide a nominal and a set of dispersed entry states at the 125 km entry interface. Two independent Entry, Descent and Landing (EDL) software programs propagated these states from entry to the surface in order to provide the 99% landing footprint on the surface. The primary EDL tool was POST, a high heritage program that was developed and used by engineers at the NASA Langley Research Center in Virginia. The backup EDL tool was AEPL, a moderate heritage program that was developed and used by engineers at the Jet Propulsion Laboratory in Pasadena California.

The entire Genesis landing footprint was intensely scrutinized both during the early planning and the final execution stages. One of the key products of the EDL team was to provide not only landing footprints for the entry capsule, but also probabilistic assessments of the possibility of landing on a person or on property outside the footprint. To minimize the chance of landing in an unwanted location, the spacecraft are targeted to miss the Earth until the final trajectory correction maneuver 48 hours before arrival. Once the final targeting maneuver has been executed, the spacecraft must be tracked for a reasonable amount of time to provide enough data to accurately predict the actual entry state, as well as the uncertainty in the entry state. The transmitters and antennas are not mounted on the Sample Return Capsules (SRC), so no tracking data is available once the SRC is separated from the primary spacecraft bus. Performing the final targeting maneuver as close to arrival as possible minimizes the effect of the maneuver execution uncertainties. Although the delta-V required to deflect the primary spacecraft so that it will miss the Earth after the entry capsule is released increases as the final targeting maneuver is moved closer to arrival, delta-V was not a consideration for either Genesis or Stardust because both spacecraft had a very large delta-V margin. The hardware designs for both vehicles significantly handicapped the maneuver design process and required careful advanced planning by the operations team to be reasonably sure that the required maneuvers could be performed due to attitude limitations.

The limiting factor for the EDL team was the operations timeline required to perform the necessary functions including: collecting tracking data, estimating the entry state and dispersions from the tracking data, propagating the entry state and dispersed states to the ground, generating landing footprints and hazard probabilities, holding a Go/NoGo conference to authorize release of the entry capsule, and uploading the commands required to make it so. Figure 3 shows a cartoon of the key events on the spacecraft and shows the location of key decision points ("buttons"). If the landing footprint predicted a dangerous overlap with a proscribed area, then release of the entry capsule would not have been authorized, and the entry capsule would have remained attached to the primary spacecraft during the divert maneuver. The "Red Button" was a set of commands that allowed release to be cancelled if new data showed that there was a problem that made release unsafe. The "Purple Button" was a set of commands that would have disabled the divert maneuver of the Genesis spacecraft bus if only a partial separation occurred. If the Genesis SRC cables had failed to cut, it would have been dangling in a position that would make the success of the divert maneuver questionable. In that case, it would be better to have the bus enter the atmosphere and burn up over the Utah Test and Training Range than to attempt a questionable maneuver that could result in reentry somewhere else. The final estimate of the landing footprint for Genesis allowed release of the entry capsule to be authorized. Separation was verified, and the SRC entered and landed close to the targeted site. The objective is the same for the return of Stardust in January.

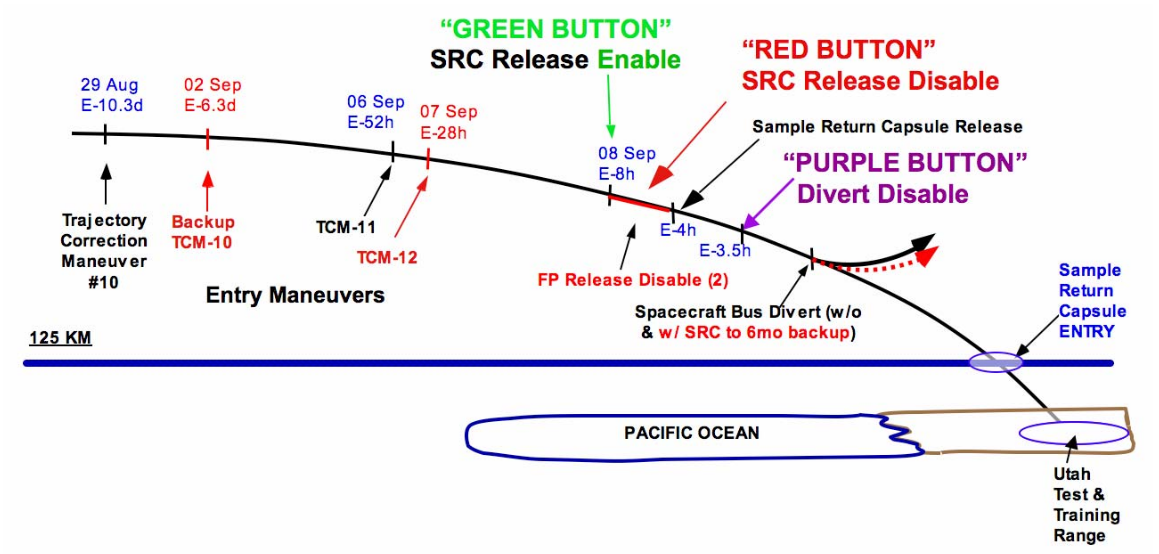

Figure 3: Cartoon Showing Genesis Timeline prior to Entry.

Table 2: Key Events Leading to Genesis Entry

Time (PDT)	Day	Event
5:00 am	Mon.	TCM-11 (Final Planned Maneuver, Sept. 6, 2004.)
5:00 am	Tues.	TCM-12 (Possible Replacement for TCM-11 if anomaly.)
9:30 pm	Tues.	Final Orbit Determination Data Cutoff
10:00 pm	Tues.	Entry State File delivered from Nav to EDL team.
11:00 pm	Tues.	Landing Site data delivery for Viewgraph Integration. 🡐
12:01 am	Wed.	Go/NoGo Team Meeting. (We decided to "Go".)
3:00 am	Wed.	Start of the Release Sequence.
3:49 am	Wed.	Begin Spin-up & turn to Release Attitude.
4:38 am	Wed.	Achieve desired Spin and Attitude for Release
4:53 am	Wed.	Release the Sample Return Capsule (SRC)
5:09 am	Wed.	Begin Divert Maneuver so S/C Bus will not reenter.
8:55 am	Wed.	Begin Entry. (Sept. 8, 2004)

The Genesis timeline was such that there was approximately 1 hour available from the time that the final Navigation entry state predicts for the Go/NoGO analysis were scheduled to be available (at 10 pm on Tuesday) until the Landing Footprints were due for incorporation into the Go/NoGo Review package. During preliminary testing of AEPL for the Genesis application, each trajectory simulation took several minutes to propagate from entry to the surface. Since 2000 such propagations were required in less than an hour wall clock time to provide the desired statistics, something had to be done to speed up the simulation throughput! Although tuning the integration tolerances and the atmospheric data table sample spacing provided some improvement, developing an automated process to distribute the cases among a collection of computer nodes so that cases could run in parallel proved to be the most effective way to achieve the required run times.

Simulation Tools:

Two independently developed trajectory propagation tools were used for Genesis landing dispersion analyses: the Program to Optimize Simulated Trajectories (POST) program[3] developed at the NASA Langley Research Center (LaRC), and the Atmospheric-Entry Powered Landing (AEPL) program developed at the Jet Propulsion Laboratory (JPL). The POST program has been developed incrementally for several decades, and has been thoroughly tested and validated for numerous projects. A program known as AEP that was used for the Mars Pathfinder entry simulation was extensively modified and upgraded for the Mars Exploration Rover program which successfully landed two rovers on Mars. This new program was renamed AEPL. The Genesis project used the same program for Earth entry analysis. Both AEPL and POST used the same LaRC-developed aerodynamics database, which provided drag and other aerodynamic coefficients as a function of Mach number and capsule angle-of-attack. Also common between the two programs were the atmospheric density models, the wind models, and the spacecraft parameters. Both programs modeled configuration changes (drogue and main parachute deployment). POST modeled “six degree of freedom” (6DOF) dynamics, from atmospheric interface to parachute deployment. From parachute deployment to landing, “three degree of freedom” (3DOF) dynamics were used, in which only the drag force is modeled and is assumed to act opposite the wind-relative velocity vector. The POST trajectory simulation seamlessly transitions from 6DOF to 3DOF. In contrast, the AEPL program used 3DOF analyses throughout. Because Genesis used unguided ballistic entries, the 3DOF results from AEPL agreed well with the 6DOF/3DOF POST simulations. Monte Carlo analyses were used to determine the landing footprint. Quantities varied in the 2000 trajectory Monte Carlo runs included entry states, atmospheric density and wind profiles, and spacecraft parachute drag coefficients. POST simulations also varied entry capsule mass moments of inertia, center of mass offsets and spacecraft parameters such as entry mass, and entry capsule drag coefficients, although these additional parameters had a negligible effect on the landing site dispersions.

AEPL:

AEPL was developed to run on a single computer. Monte Carlo cases were implemented by a “stack” of inputs that were specified in a namelist-like input file prior to starting the run. The “stacked” inputs changed specific parameters for the new case, but left all preceding parameters unchanged. Thus the input file would contain an extensive block of inputs to set up the first case, and then a set of very small blocks that only changed a few of the inputs before restarting the simulation from the entry state. (The entry state was normally one of the variables that was changed for each new case.) When running a single case, the integration parameters were normally set very tight to achieve the greatest accuracy. When running 2000 cases back to back on a single computer, the stacked run could take many hours to run on a single computer. The time required for running a large set of stacked cases for MER was reduced by relaxing the integration tolerances. We experimented with various tolerance values to find the value where we started to notice a change in the results, and then used a tolerance that was a factor of 10 smaller in order to guarantee that the results would not be noticeably different. Each order of magnitude reduction in the tolerances reduced the execution time by a factor of about 0.6. The tolerances used for the Monte Carlo cases reduced the execution time by a factor of about 0.22.

For both the Genesis and Stardust simulations, AEPL and POST used tables of densities and winds as a function of altitude. The 5000 tables of density and another 5000 tables of wind speed were generated using the EarthGRAM program for the specific arrival date and landing location. Since Genesis arrived during the morning in the fall, while Stardust will arrive in the middle of the night during winter, there are noticeable differences in the densities and the winds due to the differences in the temperature profiles and time of day. Since POST was able to perform a much more detailed simulation, including the accelerometer based parachute separation algorithm, a very fine altitude step in the atmosphere tables was used near the separation altitude in order to evaluate the possibility of "spoofing" the accelerometer measurements into triggering the parachute deployment at the wrong time. Unfortunately, the number of lines in each of the tables was more than the 2000 line limit that had been hard-coded into AEPL, so the tables had to be thinned out. Thinning out the number of values in the density and wind profiles had another benefit, because the duration of the simulation could be reduced by about a factor of 0.5 if the minimum altitude distance between samples was set to about 500 meters (about 200 samples per file.).

The largest speed improvement was enabled by the use of parallel processing. During the MER project, the conditions for all 2000 cases were specified and saved in the input-file before the first case was run. Since the input values for all the cases are independent and well known, these cases could be divided up and run in parallel on separate computers. Writing a program to generate a set of AEPL input-files was not difficult. A slightly more difficult task was to monitor the cases and then merge the finished products into a single output file. One interesting challenge was accommodating a slow computer that did not finish in the time available.

Various vendors are developing parallel processing options. Some strategies require packaging everything needed for a job together, sending it to the target computer, unpacking the job, running the job, and returning the results to the master control computer. Techniques are available for sending status messages and alerts in the event of problems. The benefit of such a system is that you can use an unlimited number of computers. All you need are accounts on multiple computers that are connected to the internet. The downside of this approach is that there are a moderate number of files, some of which are large, that have to be packaged together and transferred for each run. Some of these issues could be mitigated by installing the large ephemeris and atmospheric data files in advance. Another issue was that the primary AEPL user and scripter was not an expert in either programming or parallel processing, and was under extreme time pressure to get the system running and validated before Genesis returned to Earth. Thus the AEPL master control program was written in "quick", the programming language that the author was most comfortable using.

An array of 20 linux computers was available to members of the navigation section at JPL. Each of these computers shared a common disk array, so that any of these computers could access any of the files on the system. One of the computer nodes was used to create a set of AEPL input-files that together included all of cases and a set of shell scripts that were used to run the cases. Typically, each AEPL input-file specified a stack of 10 cases. Each input-file was saved in a different directory, where the name of the directory indicated which cases were in the input-file in that directory. For a 2000 case Monte Carlo with 10 case stacks, 200 directories would be created to contain the results of each of the 200 AEPL runs. Although it would have been possible to create 20 directories, each with a 100 case stack (i.e. where a single execution of AEPL would run on each node until all cases in the stack were completed one after the other), this approach could result in the loss of 5% of the cases for each node that was unable to finish in time. By limiting the number of cases in each run, some cases would be completed on the slowest node and available for processing by the deadline. Only those cases in a stack that had not started or was still running would be lost, rather than all the cases assigned to a very slow node. Another reason for limiting the number of cases in a stack was to enable a quick look at the results after a few cases had been run. Although looking at the results of a few cases does not give very good statistics, it can supply an early indication if there are any major problems with the predicted nominal landing location. During the testing phase, which occurred during normal business hours, some nodes could become extremely slow, depending on usage by other users. (Speed of the nodes will become more of an issue in the future as more and more users take advantage of parallel processing on multiple nodes.) The JPL navigation linux cluster is a shared resource used by nearly 300 users. One way to avoid using an extremely busy node was to test the usage level of all of the nodes before starting a Monte Carlo run. Heavily used nodes were not given any cases. A more sophisticated solution would have assigned more cases to nodes with lower usage values, but that was not done for Genesis. The time-critical cases for the Genesis final Go/NoGo decision meeting were run in the middle of the night when usage on all of the nodes was extremely light.

The design of the distribution of cases among the nodes was as simple as possible in the sense that the master process that started all of the cases running on the other nodes did not require any feedback from the other nodes. The master process started a single shell script on each of the nodes, which then processed a sequential list of AEPL runs to completion. The user could monitor progress by monitoring a list of filenames that was created when each stacked case started and when each stacked case ended. Each Monte Carlo was run in a directory whose name was representative of the data, for example, "OD154". The program would create a series of directories in "OD154", one for each stacked case. For example directory, "OD154/z_1-20/" contained all of the information needed to run a single stacked run for cases 1 to 20 out of 2000. The master program also created a shell script called "go" in each of these directories which did the following:

```
cd ~/AEPL/GENESIS/FINAL/OD154/z_1-20           # Change to this directory.
echo z_1-20 `hostname` > ../z_1-20_RUNNING     # Create Message File.
aepl.beta AEPLinputFile na > OUTPUT.temp       # Start AEPL.
mv ../z_1-20_RUNNING ../z_1-20_DONE            # Change Message Filename
# This go file was created by ... faster.      # How this file was created.
```

The user could monitor progress using the message files that were stored in the parent "OD154" directory. The UNIX shell command "ls *RUNNING" would give a list of the running cases, while "ls *DONE" would give a list of the completed cases. All of the cases were finished once there were no files with the RUNNING suffix.

After compiling a list of the available nodes, the master program also created a set of shell scripts, one for each available node. For example, the file "OD154/x_node2" was created to run the following commands on node2:

```
~/AEPL/GENESIS/FINAL/OD154/z_1-20/go            # Run first stacked set.
~/AEPL/GENESIS/FINAL/OD154/z_381-400/go         # Then run next set.
~/AEPL/GENESIS/FINAL/OD154/z_761-780/go
~/AEPL/GENESIS/FINAL/OD154/z_1141-1160/go
~/AEPL/GENESIS/FINAL/OD154/z_1521-1540/go
~/AEPL/GENESIS/FINAL/OD154/z_1901-1920/go       # The last set for node2
```

Finally, the master program would create and then execute the following shell script that started running cases on each of the nodes:

```
echo Starting ... x_node2
ssh node2   sh ~/AEPL/GENESIS/FINAL/OD154/x_node2 &

echo Starting ... x_node3
ssh node3   sh ~/AEPL/GENESIS/FINAL/OD154/x_node3 &

echo Starting ... x_node4
ssh node4   sh ~/AEPL/GENESIS/FINAL/OD154/x_node4 &
…
```

The final latitude and longitude results were merged together in "quick" and written to a file in the parent directory. This file was then processed by the earthLS program as described later in this paper.

A team of experts are developing more sophisticated process that should be able to reduce the total run time required. The new process assigns cases dynamically. A new set of cases is assigned to a node only after the previous set of cases has finished running. For the Genesis mission, the association between the cases and the nodes was made before any cases were run, so the time required to complete all 2000 cases was determined by the slowest node. Although this approach of assigning all the cases at the start worked well when all of the nodes were running at about the same speed, a better way would have been to assign a new stacked case to a node after it had finished running the previous stacked case. Dynamically assigning the cases can minimize the total time required because fewer cases would be run on the slower nodes. The master control program to dynamically assign the cases would be more complex than that used for Genesis, because it would have to actively monitor the status of the cases in order to start the next case in a timely manner. Ten new nodes have recently been added to the cluster. The new nodes are significantly faster than the older nodes. In order to minimize the throughput time for Stardust, the assignment of cases should be modified so that the faster computers will be assigned more cases. The current strategy of assigning about the same number of cases to each node does not take advantage of the speed and usage differences between the nodes.

Security is another important consideration. JPL rules require password authentication in order to log onto a computer. Although the user ID and password are the same for all of the nodes, each node is a separate computer, and requires authentication before a user can access that node, even when coming from another node on the cluster. Since it is against the rules to store the password in an electronic file on the computer, some other means was required to enable a process on one node to start processes on the other nodes. The technique used a feature of OpenSSH to enable remote connections without requiring a password for each and every connection. OpenSSH authentication uses RSA and DSA protocols that are based on complementary numerical keys. Some setup is required prior to the first use. After the user has logged onto the node that will be the master node, three lines must be entered: "ssh-agent csh", "ssh-add", and a unique "passphrase" that the user has set up previously. From that point on, processes can be started on other nodes without requiring the user to type in a password each time. Further information about OpenSSH is readily available on the web.

POST:

POST was also developed to run on a single computer by engineers at the NASA Langley Research Center. POST has been thoroughly tested and used successfully on numerous projects, including MER and Genesis. Because it simulates the full 6 Degree of Freedom rigid body motion of the Sample Return Capsule prior to parachute deployment, it must perform even more calculations than AEPL. In order to reduce the total execution time, NASA Langley engineers developed UNIX scripts to run POST simultaneously on multiple processors. The approach was very similar to that used to run AEPL. Like AEPL, the core POST program was not changed in any way. Unlike AEPL, POST was run on a dedicated 64 node SGI Origins cluster. A dedicated cluster was required because POST was the "prime" analysis tool during operations, and AEPL was the backup. Since they were run on different clusters, there was no competition for resources. Like AEPL, UNIX scripts were used to drive the processing. Unlike AEPL, which used a program to accumulate the results once all the cases had finished, the POST process used a unix "cat" command to concatenate partial results every cycle. Unlike AEPL, the processors were not run independently. Each of the 64 processors in the cluster ran one case to completion. When all 64 cases were complete, those results were concatenated with the previous results in a single file. Then the next 64 cases were run to completion and those results were concatenated into the single results file. This process was continued until all 2000 cases were run. Since each processor was dedicated to running a single case, and since each case took about the same time to run, there was no penalty associated with waiting for all 64 processors to finish before starting the next set of runs. Since the SGI Origins cluster was designed to run multiple cases in parallel like this, standard tools were available to help the developer build and run the system.

The Genesis Final Product:

AEPL and POST each produced a file containing 2000 pairs of latitudes and longitudes that represented a possible landing location. Both of these files had to be processed by a visualization tool called EarthLS[4], a slightly modified version of a program called MarsLS that was developed for the MER project[5]. EarthLS plots the latititude-longitude points on a calibrated map and then computes statistics associated with those points, such as the Gaussian ellipse that encloses 99% of the points. To avoid excessive clutter on the image, the points are hidden once the ellipses have been computed. Figure 4 is the final EarthLS graphic shown at the Genesis Go/NoGo decision meeting. It shows the 99% Gaussian ellipses for both AEPL and POST for the last 4 Orbit Determination solutions (OD151 – OD 154) prior to separation. POST and AEPL agreed so well with each other, that you can not tell the difference in the figure. Even more important is the fact that the 4 different Navigation solutions also agree such that you can barely tell the difference between the 4 ellipses (Green, Purple, White, and Red). The 99% Ellipse Footprint for OD154 had a major axis of 41.9 km, a minor axis of 27.1 km, and an azimuth of 137.24 deg (E. of North). The centers of the final ellipses are slightly northeast of the Genesis Target, which is plotted as an Orange "+" at 40.02° N (geocentric) latitude, 246.48° East longitude. The smaller Orange ellipse, which is slightly larger than the final 99% ellipse, is the expected 99% ellipse centered on the target. The very large orange ellipse represents the 6-sigma boundary that is also centered on the target. If the final predicted landing location had been outside of this 6-sigma boundary, then the Go/NoGo decision would have been an automatic "No" because something unexplained would have to be affecting the trajectory for it to be that far from the target. If the final predicted landing location had been between the 3-sigma and 6-sigma ellipses, then the project manager would have to be convinced that there was a good and well understood reason why the offset was more than 3-sigma. Since the final predicted landing location was

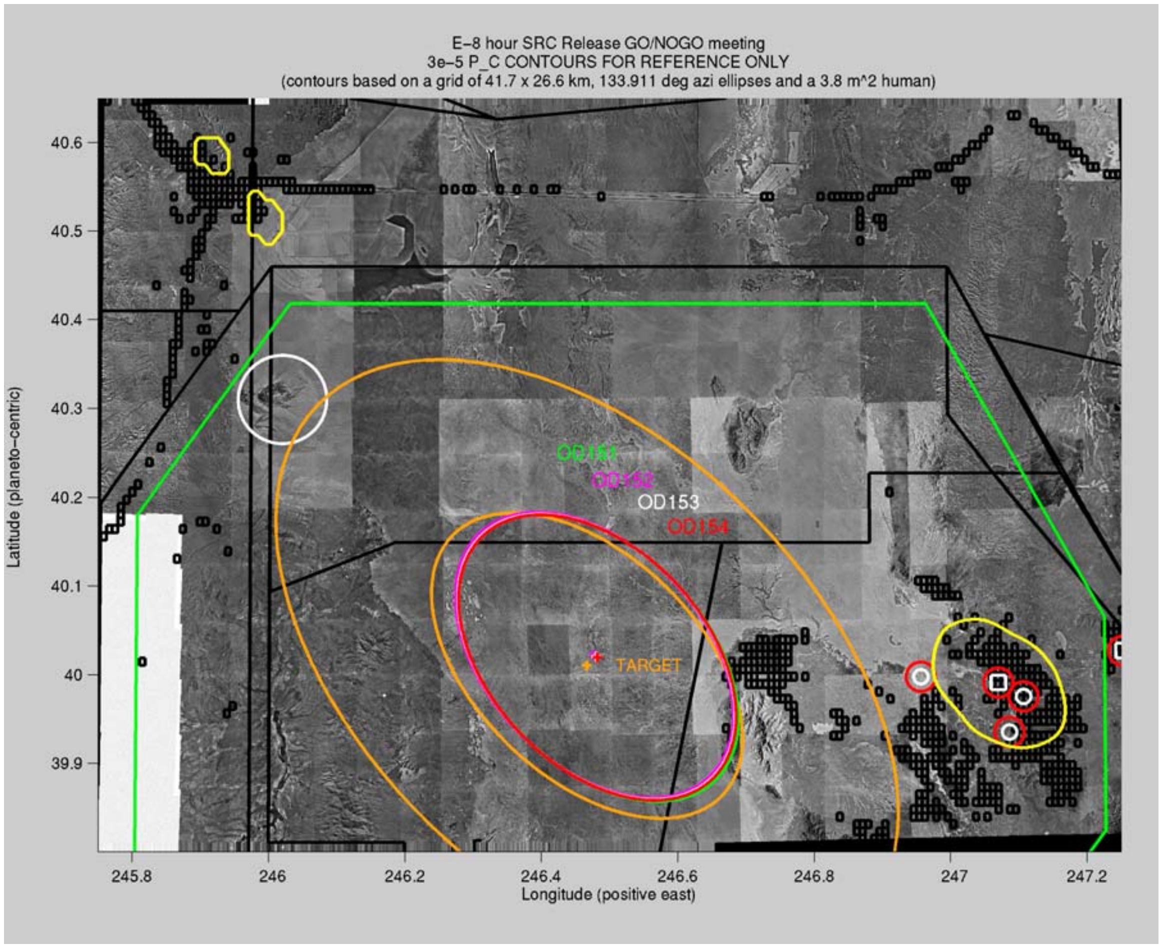
E–8 hour SRC Release GO/NOGO meeting
3e–5 P_C CONTOURS FOR REFERENCE ONLY
(contours based on a grid of 41.7 x 26.6 km, 133.911 deg azi ellipses and a 3.8 m^2 human)
Latitude (planeto–centric)
40.6
40.5
40.4
40.3
40.2
40.1
40
39.9
OD151
OD152
OD153
OD154
TARGET
245.8
246
246.2
246.4
246.6
246.8
247
247.2
Longitude (positive east)

Figure 4: Final EarthLS graphic shown at the Genesis Go/NoGo decision meeting.

extremely close to the target, and since the size of the dispersion was very close to the expected value, the decision to release the SRC depended on the status of the spacecraft hardware that was inferred from telemetry. The Genesis project manager decided to release the spacecraft, which indeed landed very close to the target.

Some of the other features shown in Figure 4 are the following. The bright green set of straight lines represents the “UTTR-constraint” that is located 2.5 nautical miles inside the actual UTTR boundary. This virtual “Green Fence” represented the actual constraint on the landing location. The white circle near the upper left corner identifies a geographic hazard, while the yellow contours represent keepout zones to minimize the risk to humans. The Red circles near the lower right corner represent 1 nautical mile keepout zones around Dugway facilities. If the final predicted landing location (i.e. the center of the ellipse) was inside of one of these red circles, release could not be authorized. If the final predicted landing location was inside the yellow contour, then the probability of injuring a human would have exceeded the requirement, and release would not have been authorized. The background is a map of the Utah Test and Training Range.

EarthLS is also used to compute the probability of violating various requirements associated with landing in hazardous or populated areas. Table 3 shows the EarthLS summary table that was produced by EarthLS and reviewed at the Genesis Go/NoGo meeting just prior to the authorization to release the sample return capsule. All of the safety requirements were met.

Table 3: Probability Table & Requirements Checklist from Genesis Go/NoGo meeting.

NAVIGATION DECISION FACTORS:	CRITERION	POST	AEPL	VIOLATIONS POST	VIOLATIONS AEPL
Impact points NOT in Nav Delivery Zone	< 1e-02				
Nominal Ellipse:		0.00e+00	0.00e+00	0	0
14km Offset Ellipse:		0.00e+00	0.00e+00	0	0
IPs meet NASA Pc for Public Individual	< 1e-06				
LandScan-Population Data:		3.33e-10	3.02e-10	0	0
UTTR-Population Data:		7.84e-10	9.59e-10	0	0
IPs meet NASA Pc for Public Collective	< 3e-05				
LandScan-Population Data:		5.60e-09	6.24e-09	0	0
UTTR-Population Data:		4.18e-09	5.11e-09	0	0
IPs meet NASA Pc for Mission Individual	< 1e-05				
UTTR-Population Data:		2.35e-10	2.89e-10	0	0
IPs meet NASA Pc for Mission Collective	< 3e-04				
UTTR-Population Data:		1.65e-09	2.03e-09	0	0
IPs meet UTTR Pc for Public Individual	< 1e-07				
UTTR-Population Data:		7.84e-10	9.59e-10	0	0
IPs meet UTTR Pc for Public Collective	< 3e-05				
UTTR-Population Data:		4.18e-09	5.11e-09	0	0
IPs meet UTTR Pc for Mission Individual	< 3e-06				
UTTR-Population Data:		2.35e-10	2.89e-10	0	0
IPs meet UTTR Pc for Mission Collective	< 3e-04				
UTTR-Population Data:		1.65e-09	2.03e-09	0	0
Nominal points enter Dugway keep-out zone	0			0	0
IPs enter UTTR-provided areas (BL,FS,DPG,etc):	N/A	2.00e-02	2.45e-02	N/A	N/A
NW IPs enter LandScan areas (Wendover,I-80):	N/A	4.24e-06	5.07e-06	N/A	N/A
NUMBER OF APPLICABLE CRITERIA VIOLATED ABOVE:				0	0

Valid Navigation Solutions	CRITERION	OD	MNVR	POST	AEPL
Concurrence from Navigation Advisory Group	1	1	1	1	1

Conclusions:

Parallel processing proved to be a very effective way to achieve the simulation throughput required to support operations for Genesis. Very similar techniques were used to speed up both AEPL and POST by running multiple instances of each program on multiple processors. The results from the multiple runs had to be merged together to create a pair of input files that were processed together by EarthLS. POST had the advantage of having a larger number of processors dedicated to Genesis during the final critical operations. Both programs were able to deliver the required results in the time that was allocated. These same techniques will be used to support operations for Stardust in January of 2006.

Acknowledgements:

The work described was performed at the Jet Propulsion Laboratory (JPL), California Institute of Technology and the NASA Langley Research Center under contract with National Aeronautics and Space Administration (NASA). The Genesis and Stardust spacecraft were built by Lockheed Martin Astronautics (LMA) in Denver Colorado. Tracking support was provided by the Deep Space Network. Both spacecraft are managed by JPL and are operated by joint teams with members from JPL, LMA and NASA Langley.

References:

[1] Martin W. Lo, Bobby G. Williams, Williard E. Bollman, Dongsuk Han, Yungsun Hahn, Julia L. Bell, Edward A. Hirst, Robert A. Corwin, Philip E. Hong, Kathleen C. Howell, "Genesis Mission Design", AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Boston, MA, Aug. 10-12, 1998. AIAA-1998-4468

[2] Edward A. Hirst and Chen-Wan L. Yen, "Effect of unbalanced attitude control burns on STARDUST trajectory design"**,** AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Boston, MA, Aug. 10-12, 1998. AIAA-1998-4189

[3] Brauer, G. L., Cornick, D. E., and Stevenson, R., "Capabilities and Applications of the Program to Optimize Simulated Trajectories (POST)," NASA CR-2770, Feb. 1977.

[4] Wawrzyniak, G. G., "MarsLS User Guide Version 2.1", JPL D-24497, 11 Nov. 2003.

[5] Philip C. Knocke, Geoffrey G. Wawrzyniak, Brian M. Kennedy, Prasun N. Desai, Timothy J. Parker, Matthew P. Golombek, Thomas C. Duxbury, and David M. Kass, "Mars Exploration Rovers Landing Dispersion Analysis", AIAA/AAS Astrodynamics Specialist Conference and Exhibit 16 - 19 August 2004, Providence, Rhode Island. AIAA 2004-5093